Random Graphs (Volume 2). Edited by Alan Frieze and Tomasz Łuczak. John Wiley & Sons, Inc., New York, Chichester, Brisbane, Toronto, and Singapore. (1992). 285 pages. $98.00.

Contents:

1. Probability distributions related to the local structure of a random mapping. (Sven Berg and Jerzy Jaworski). 2. Maximum cuts and quasirandom graphs. (F.R.K. Chung and R.L. Graham). 3. Inequalities for random walks on trees. (Luc Devroye and Amine Sbihi). 4. Spanning trees in random graphs. (Peter Dolan). 5. Subgraphs of large minimal degree. (Paul Erdős, Tomasz Łuczak, and Joel Spencer). 6. On small subgraphs of random graphs. (Alan Frieze). 7. On the number of vertices in the complete graph with a given vertex as nearest neighbor. (Lars Holst). 8. When is a graphical sequence stable? (Mark Jerrum, Brendan McKay and Alistair Sinclair). 9, On the stack ramifications of binary trees. (R. Kemp). 10. The number of permutations with cycle lengths from a fixed set. (V.F. Kolchin). 11. On the rank of a random submatroid of projective geometry. (Wojciech Kordecki). 12. Sparse random graphs with a given degree sequence. (Tomasz Łuczak). 13. A law of large numbers for path lengths in search trees. (Hosam M. Mahmoud). 14. On random mappings of finite sets. (Ljuben R. Mutafchiev). 15. Proving normality in combinatorics. (Andrzej Ruciński). 16. Remarks on the stochastic travelling salesman. (Eli Shamir). 17. Random signed graphs with an application to topological graph theory. (Martin Škoviera). 18. A shape result for first-passage percolation on the Voronoi tessellation and Delaunay triangulation. (Mohammad Q. Vahidi-Asl and John C. Wierman). 19. On components of random subgraphs of the n-cube. (Karl Weber).

Images of Programming: Dedicated to the Memory of A.P. Ershov. Edited by D. Bjørner and V. Kotov. North-Holland, Amsterdam, London, New York and Tokyo. (1991). 270 pages. $97.50.

Contents:

I. Ershov and informatics. 1. Andrej Petrovich Ershov. (I.V. Pottosin). 2. Selected biography of Ershov's papers. (I.V. Pottosin). 3. Andrei P. Ershov—a friend and a rival. (S.S. Lavrov). 4. "From hidden places knowledge I obtained." (K. Levitin). 5. Maturing of informatics. (J. Gruska and H. Jürgensen). 6. Quality of education with application of new information technologies. (B. Sendov and A. Eskenasi). II. Computing science. 1. Theory and practice. (D.E. Knuth). 2. Analysis of program optimization possibilities and further development. (I.V. Pottosin). 3. Transformational approach to program concretization. (V. Kasyanov). 4. From partial evaluation to mixed computation. (M.A. Bulyonkov). 5. Essence of generalized partial computation. (Y. Futamura, K. Nogi and A. Takano). 6. An algebra of mixed computation. (V.E. Itkin). 7. Static semantics, types, and binding time analysis. (N.D. Jones). 8. Prescribing behaviours. (W.M. Turski). 9. Four domains for concurrency. (J.W. de Bakker and J.H.A. Warmerdam). 10. An algebra of concurrent non-deterministic processes. (L.A. Cherkasova and V.E. Kotov). 11. Connectedness and synchronization. (A. Mazurkiewicz, A. Rabinovich and B.A. Trakhtenbrot). 12. Higher order dataflow schemas. (E. Tyugu). 13. Rapid constructions of algebraic axioms from samples. (J.M. Barzdin and G.J. Barzdin). 14. On conservative extensions of syntax in system development. (A. Blikle, W. Tarlecki and M. Thorup). 15. A theory for the derivation of combinational C-mos circuit designs. (C.A.R. Hoare). 16. A bridge between constructive logic and computer programming. (N.N. Nepejvoda).

Software Engineering: A Holistic View. By Bruce I. Blum. Oxford University Press, New York and Oxford. (1992). 588 pages. $49.95.

Contents:

1. The software process. 1.1 What is software engineering? 1.1.1 The origins of software engineering. 1.1.2 Some modern views of software engineering. 1.2 Models of the software process. 1.2.1 A traditional waterfall model. 1.2.2 The essential software process. 1.2.3 Some alternative models. 1.3 Software engineering as problem solving. 1.3.1 Problem solving. 1.3.2 Modeling concepts. 1.3.3 Methods, tools, and environments. 1.4 Facts, myths, and perceptions. 1.4.1 The famous GAO report. 1.4.2 Software engineering findings. 1.5 Some final comments. 2. Requirements analysis and modeling. 2.1 Approaching requirements analysis. 2.1.1 Distinguishing between the what and how. 2.1.2 The role of prototypes. 2.2 The configuration management case study. 2.3 Modeling techniques. 2.3.1 Data models. 2.3.2 Process descriptions. 2.3.3 Formal methods. 2.3.4 Textual specifications. 2.4 Implicit requirements. 2.4.1 The user interface. 2.4.2 The ilities. 2.5 The next steps. 3. Modeling-in-the-large. 3.1 Decomposition and composition. 3.2 Structured analysis and structured design. 3.2.1 Structured analysis and the SCM case study. 3.2.2 Structured design. 3.2.3 Variations on structured analysis. 3.3 Jackson system development and programming. 3.3.1 Jackson program design (JSP). 3.3.2 Jackson system development (JSD). 3.4 Comparisons and alternatives. 4. Modeling-in-the-small. 4.1 Implementing the system. 4.1.1 Prerequisites to implementation. 4.1.2 Elements of style. 4.1.3 Views of the process. 4.2 Encapsulation techniques. 4.2.1 Some foundations of abstraction. 4.2.2 Abstract data types. 4.2.3 Object-oriented programming. 4.2.4 Object-oriented design and analysis. 4.3 The program proof. 4.4 Concluding observations on modeling. 5. Verification and validation. 5.1 On the importance of being ernest. 5.2 Before the programs exist. 5.3 After the programs exist. 5.3.1 Taxonomy of testing. 5.3.2 Static analysis and complexity analysis. 5.3.3 White box testing. 5.3.4 Black box testing. 5.3.5 Integration. 5.4 The cleanroom. 5.5 A summing up. 6. Managing the process. 6.1 Overview of management. 6.2 Principles of software project management. 6.2.1 Tools for technical management. 6.2.2 The risk management process. 6.2.3 Project organization to reduce risks. 6.2.4 Building a management model. 6.2.5 Process improvement. 6.3 Process improvement technologies. 6.3.1 Reuse. 6.3.2 Reengineering. 6.3.3 Automated support. 6.4 Some final observations.